Steganography in Arabic Text Using Full Diacritics Text

Ammar Odeh
Computer Science & Engineering,
University of Bridgeport,
Bridgeport, CT06604, USA
Aodeh@bridgeport.edu

Khaled Elleithy
Computer Science & Engineering,
University of Bridgeport,
Bridgeport, CT06604, USA
elleithy@bridgeport.edu

Abstract

The need for secure communications has significantly increased with the explosive growth of the internet and mobile communications. The usage of text documents has doubled several times over the past years especially with mobile devices. In this paper we propose a new Steganogaphy algorithm for Arabic text. The algorithm employs some Arabic language characteristics, which represent as small vowel letters. Arabic Diacritics is an optional property for any text and usually is not popularly used. Many algorithms tried to employ this property to hide data in Arabic text. In our method, we use this property to hide data and reduce the probability of suspicions hiding. Our approach uses a performance metric involves the file size before and after adding Diacritics and ability to hide data with being suspicious.

Keywords: Kashida, Carrier file, Zero width chracter, Information Hiding, Diacritics.

1. INTRODUCTION

1.1. Background

Steganogaphy is a combination of two Greek words. Where "Stegano" means hidden and “Graptos" means writing. The secure data is embedded into other object, so a middle attacker can't catch it [1]. Invisible ink is an example for Steganogaphy where a readable message is transferred between source and destination. Any intruder in the middle can read the message without knowing about the hidden data. Meanwhile, authorized persons can read it depending on substances features [2], [3].

Ancient Greece used to shave the messenger head and then wait until hair grows back, then the message can be sent to the destination [1]. Depending on this method, we have 2 possibilities:-

1. If the message arrives, the receiver can read the message and recognize if the message changed or not.
2. If the message did not arrive, this mean an attacker was able to detect the message.

1.2. Motivation

Steganogaphy algorithms depend on three techniques to embed hidden data in carrier files.

1. Substitution: Exchange some small part of the carrier file by hidden message. Where middle attacker can't observe the changes in the carrier file. On the other hand, choosing replacement process it is very important to avoid any suspicion. This means to select insignificant part from the file then replace it. For instance, if a carrier file is an image (RGB) then the least significant bit (LSB) will be used as exchange bit [4].
2. Injection: By adding hidden data into the carrier file, where the file size will increase and this will increase the probability being discovered. The main goal in this approach is how to present techniques to add hidden data and to void attacker suspicion [4].

Propagation: in this approach there is no need for cover object. It depends on generation engine which is fed by input (hidden data) to produce a mimic file (graphic or music or text document).

Steganogaphy process consists of three main components as show in Figure 1.

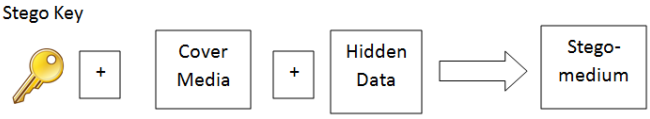


Figure 1. General components of Steganogaphy

Different types of cover media can be used in Steganogaphy including image, sound, video and text. Choosing a carrier file is very sensitive where it plays a key role to protect the embedded message. Successful Steganogaphy depends on avoiding suspicion. Stegoanalysis starts checking a transmitted file if there is any suspicion and this defeats the main goal of Steganogaphy [3][4].

Text Steganogaphy represents the most difficult type, where lack of data redundancy in text file in comparison with other carrier files [5], which reduces the capacity of the hidden data. Furthermore, text Steganogaphy depends on language used, where each language has some characteristics which is completely different from others. For example, letter shape in English language doesn’t depend on its position in the word, where Persian/Arabic letters have different forms depending on their position in the word [6].

The algorithm presented in this paper aims to hide text inside text by employing Arabic language and applying a

random algorithm to distribute the hidden bits inside the message. The main reasons for choosing the Arabic language are:

1. The proposed algorithm depends on multi dotted points letters. Therefore, the algorithm must employ a language that has as much as possible of dotted letters. For example, the Arabic Language has 5 multipoint letters; the Persian language has 7 letters [7], where the English language does not have any one.
2. Wealth availability of electronic textual information.
3. There is a little research on other languages compared to English.
4. Can be extended to other languages like Urdu, Farsi and Kurdish.

1.3. Main Contributions and Paper Organization

An efficient algorithm is presented in this paper. The main idea is to use Kashida, Zero width characters in Arabic that enable us to hide two bits per one letter. Most of the pervious algorithms hide one bit for one letter. In addition we will use parallel connection, randomization strategy to avoid any suspicions.

The rest of this paper is organized as follows. In section II we discuss some text Steganography techniques. Employing Kashida and Zero width hidden algorithms are discussed in section III. Finally, conclusion remarks are offered in section IV.

2. PRIOR WORK

Text Steganography is divided into two categories. The first one is semantic based while the second is formatting based as shown in Table I. In this section we present some of these examples. In Table I we present simple comparison between semantic and formatting methods.

Table I. Comparing between texts Steganography

	Semantic Method	Format Method
Amount of hidden data	Small amount	More than semantic
Flaws	Sentence meaning	notice from OCR or retyping

A Steganography evaluation criterion depends on the amount of data that can be hidden and the main problem faces the method.

In this section we compare ten algorithms to hide data inside text documents. The last two of them deal with Arabic and Persian languages.

1. Word Synonym

Word Synonym is classified as semantic method and it depends on replacing some words by their synonym. This technique will convey data without any suspicion. Meanwhile, the hidden data is small relative to other methods. Moreover it may change the sentence meaning [7][10][11].

2. Punctuation method

Using punctuations like (.) and (;) to represent hidden text for example, “NY, CT, and NJ" is similar to "NY, CT and NJ" where the extra comma is used to represent 1 or to represent 0. The amount of hidden data in this method is very small in comparison to the amount of cover media. Inconsistence use of punctuation will be noticeable from a Stegoanalysis perspective [9].

Table II. Using Word Synonym

Word	Synonym
Big	Large
Find	Observe
Familiar	Popular
Dissertation	Thesis
Chilly	Cool

3. Line Shifting

Line shifting involves vertically shifting a line a little bit to hide information to create a unique shape of the text. Unfortunately line shifting can be detected by character recognition programs. Moreover, retyping the document will remove all hidden data. In Figure 2 we present some example about line shifting where the vertical shifting will be very small (1/300 inch) so in normal case no one can notice it.

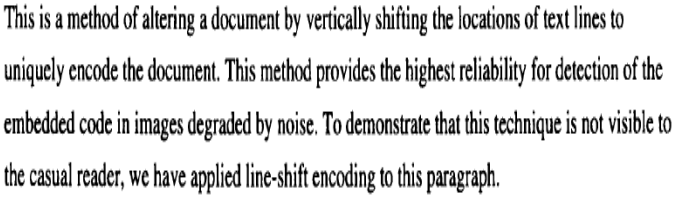


Figure 2. Line shifting where second line is shifted up 1/300 inch [7]

4. Word Shifting

In this method, changing space between words will enable us to hide information. Word shifting will be noticeable by Optical Character Recognition (OCR) by detecting space sequence between words.

5. SMS Abbreviations

Recently most transmitted SMS messages are using abbreviations for simplicity and secure communications in different applications such as internet chatting, email, and mobile messaging.

The main advantage of this method is to save the time of writing and the space needed to write messages and manipulated keyboard limitation character.

Certain algorithms use some numbers to convey specific information. As mentioned above SMS abbreviations can be used in specific applications. In case these abbreviations are not used in the common applications, suspicion from Stegoanalysis systems might be raised.

Table III. Some SMS Abbreviations

Abbreviation	Meaning
ADR	Address
ABT	About
URW	You are welcome
ILY	I love you
EOL	End of lecture
AYS	Are you serious?

6. Text Abbreviations

Text abbreviations are similar to SMS abbreviations, where a dictionary is created between each word abbreviation and its meaning. The abbreviations dictionary is published to the communicating parties. Abbreviation represents one method to hide data. For example if you send "see" that is meant to be "do you understand"[12].

7. HTML Spam Text

This method depends on HTML pages, where their tags and their members are insensitive. For example <BR> is equal to
 , <br> and <bR>. The hidden data depends on letter case upper or lower to embedded 0 or 1.

8. TeX ligatures

Some special groups of letters are joined together to create single glyph as shown in figure 3. The algorithm finds the availability ligature in the text to hide a bit in each one. For example if we want to hide 1 we write fi to f {}i which creates some space between f and i. Otherwise we encode 0 [5].

The same algorithm can be applied to Arabic character "ﻉ" or " ﻠ". This algorithm has two problems. The first one is that the file size increases when we apply extension in our text. The second problem is that if OCR notes the font change, it can be easy recognize the hidden message [6][5].

9. Vertical displacement of the points

This algorithm performs well if it is applied in pointed letter. The English language has only {i, j} as pointed letters In contrast, the Arabic and Persian languages have pointed character sets (Arabic has 26 letters which 13 of them are pointed, Persian has 32 letters which 22 of them are pointed). The algorithm encodes 1 to shift up the point otherwise encode 0. This method can encode a huge number of bits, and need a strong OCR to recognize the changes. In contrast retyping will remove the entire message [7].

10. Arabic Diacritics

Arabic language uses different marks to distinguish between words that have same letters. it depends on Arabic Diacritics (Harakat), where Diacritics are optional.

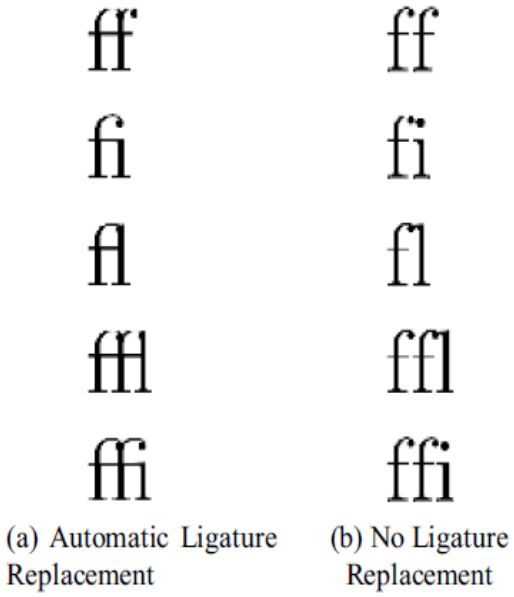


Figure 3 .Join between characters [5]

Figure 4. Vertical shifting point [7]

Most of Arabic scripts can be read without Diacritics which depends on language grammar. The most occurrence is Fatha " ◌َ " which can be used to encode 1 otherwise encode 0.This algorithm enhances the reuse of cover media. Furthermore, the carrier file size is reduced depends on hidden message. Meanwhile, ORC scans the message with different diacritics can conclude that there is hidden data, in addition retyping will remove the embedded message [8].

Other research efforts studied adding extra diacritics to text like [11], and to increase robustness of data used other scenario to hide data in image.

In [17], the algorithm hides data by showing the diacritics if it encodes 1, otherwise it hides the diacritics. The main disadvantage of this approach that it might be observed if it is compared more than once with the original text.

11. Using the Extension 'Kashida' Character

The strategy used in this method depends on letter extension (Kashida). Kashida cannot be added at the beginning and at end of word but it can be added between letters in words. In other words, un-pointed letter with extension is used to hide zero and pointed letter with extension will hold 1. [13] In this approach, message content will not be affected. On other hand, a new Unicode will be added (0640).

Table IV. Some Letters with mark and their Pronunciation

Haraka	Letter with Haraka	Pronunciation
Dama	دُ	Do
Kasra	دِ	De
Fatha	دَ	Da

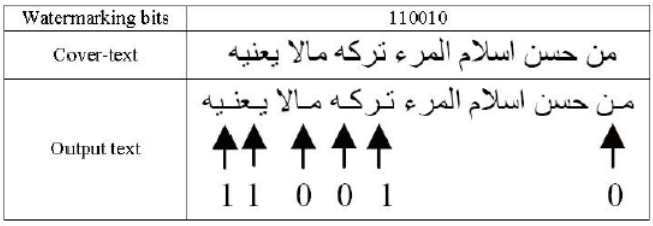


Figure 5. Kashida character after pointed letter [14]

As shown in Figure 5, not all characters can hide a bit. Therefore, Stegoanalysis may suspect the message and discover the embedded content.

12. Using the Extension 'Kashida' Character

This algorithm uses one Kashida to represent zero and 2 Kashida represent one. Since each letter needs 16 bits to be represented, the algorithm use-mapping table to which each character is mapped. So instead 16 bits, we can represent each letter by 6 bits only and this will save 10 bits. [15]

13. Using Pseudo-Space and Pseudo Connection characters

This technique is also called zero width non-joins (ZWNJ) and zero width joiner (ZWJ) characters. The algorithm classifies letters to join or non-join letters. If it is required to hide 1 zero width is added, otherwise hide 0 is hided. [16]

3. PROPOSED ALGORITHM

Some of the Arabic characters features support different Steganogaphy algorithms. In The following section, we explore some Arabic languages properties to use some of its attributes to hide large amount of data inside Unicode file (Arabic language).

1. Writing Direction:-

Arabic text written from right to left. It is a unidirectional language and numbers are read and written in the same direction.

2. Letter connectivity:-

Most of the Arabic letters in the word are connected with the previous letter and next one. Therefore, the letter may have different shapes depending on its position in the word as shown in Figure 6.

3. Dotted letters

Some Arabic letters have one, two and sometimes three points. These points affects the letter's pronunciation as Table V shows.[18]

Figure 6. Different (Mem) Letter shapes depend in its position in the word.

4. Diacritic Signs (Harakat)

Arabic language has nine Diacritics as Table VI shows. Usually those Diacritic optionally appears in most Arabic text.

The main technique used in our algorithm is to employ optional properties of Arabic language, which is the Diacritic. We will be applying vertical shifting of Diacritic relatively to the character. Zero is represented by no change and 1 is represented by increasing the distance between the letter and its dicraties as shown in Table VII.

Table V: Some pointed letter in Arabic letter

Letter	Pronounce
ب	B
ت	T
ث	Th

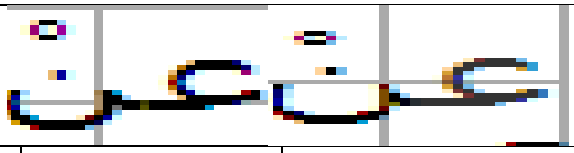
Table VI: Arabic Diacritics

Name of Diacritic	Shape of Diacritic
Dhammah	ـُ
Sukkon	ـْ
Fatha	ـَ
Kasrah	ـِ
Shaddah	ـّ
Tanween Fatha	ـً
Tanween Dham	ـٌ
Mad	~
Tanween Kasr	ـٍ

Therefore, using this optional property in the text enables us to pass a large amount of data by using small cover media (text). Usually Diacritic does not have a standard distance or position with respect to the letter Therefore, if we apply

vertical shifting by 1/200 inches it will not be noticeable by any Stegoanalysis tools. Moreover, the text size will not be largely affected.

Table VII Vertical shifting of **Diacritic**

Word		
Hidden Data	1	0

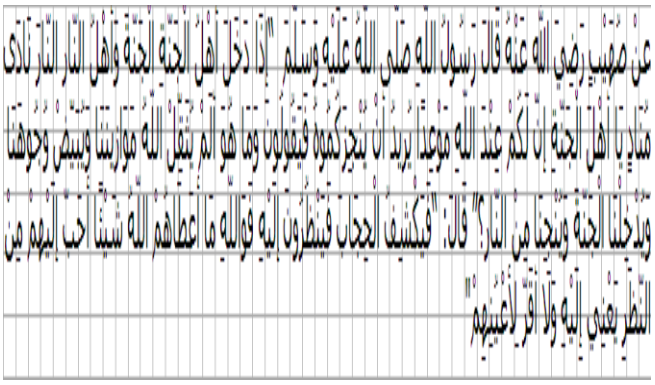


Figure 7. Grid systems for none changing Arabic text

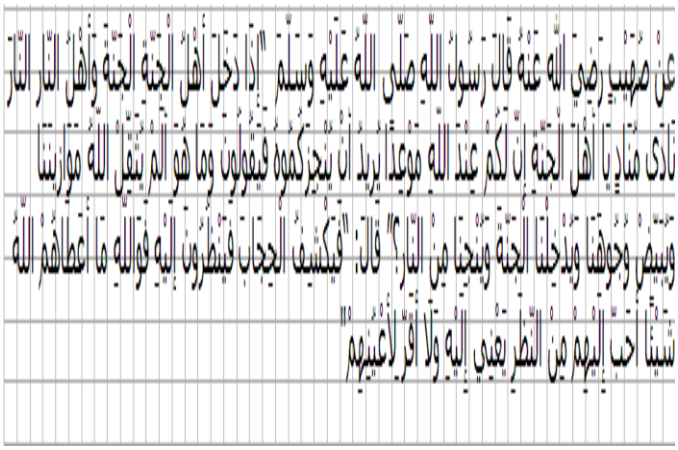


Figure 8. Grid systems for vertical shifting

Algorithm I Data Hidden

Read text from a website , Hide message from user
While !EOF // not end of file
Read letter and check
If letter has Diacritic then ==True
{// start if statement
Read bit from hidden message
If Hidden bit ==1 then
Shift up Diacritic
Else
Nochange
}/end if statement
Insert letter to Outputfile//stegotext
Next Letter
End While
Output Segotext

Algorithm II Data Extracting

Read text from Segotext
While !EOF // not end of file
Read letter and check
If letter has Diacritic then ==True
{// start if statement
If Shift up Diacritic == True
then
Insert 1 to output text
Else
Insert 1 to output text
}/end if statement
Next Letter
End While
Output Hidden data

4. DISCUSSION AND ANALYSIS

We applied our algorithm into eight websites as table XX shows
In Table VIII we analyze the website by applying different matrices.
1. File size:
Table VIII displays the size of the websites with and without Diacritic and the difference between these two cases. Most of websites show a change in their size less than 1.4 KB and a ratio less than 7% of the website size.
2. Capacity ratio:
To represent the percent of the number of bits that can be hidden in a specific file size, we use:

Capacity ratio =Number of bits can be hidden/ carrier file size

Figure 9 shows relation of number of website data can be hidden inside it.

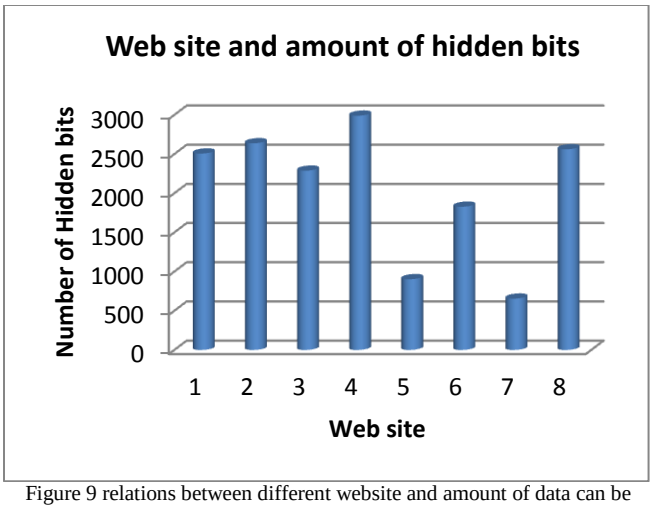


Figure 9 relations between different website and amount of data can be hidden inside it.

Where data hidden inside page related to size of that page, as figure 10 shows the difference between diacritic text and none Diacritic text file size.

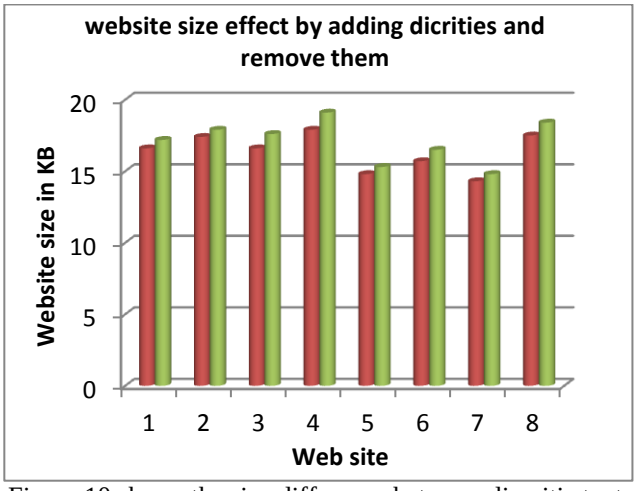


Figure 10 shows the size difference between diacritic text and none Diacritic text

The results of the experiments performed illustrate that the presented algorithm has the following advantages:

1. Size of cover media (text file) is minimally affected as shown in Figure 10.
2. Large amount of data can be hidden inside the cover media as Figure 9 shown
3. File format not effected in abnormal
4. About two billion people use Arabic language throughout the world
5. Can be extended to other similar languages like Pashto, Urdu, and Persian.

The main disadvantage of the presented algorithm is that nowadays most of the Arabic text did not include diacritics. Using diacritics may lead to some suspicions on carrier file.

5. Conclusion

Hiding data in different cover media represent one of challenging security issues. One of the difficult media to use for hiding data is a text, where embedding data may affect the text format. The file size and format change will increase the probability of being discovered using Stegoanalysis tools and this will lead to reveal the hidden data. The algorithm presented in this paper use ones of Arabic language property, which is called diacritic, without effect on the file size or text format in any abnormal or suspicious way. Comparison of this algorithm with other techniques in the same categories [8][11], our algorithm does not remove any of the text diacritics and this will reduce the probability suspicion by Stegoanalysis tools. Furthermore, vertical shifting will not affect the file size in any noticeable way.

Table VIII Vertical shifting of Diacritics

	Websites	Number of letter	Size without	Number of Diacritic	Size without Diacritic	Size with Diacritic	change in size	Size Ratio	Capacity Ratio (Bit/Kilo byte)
1	http://mubashermisr.aljazeera.net/	6138	3631	2507	16.6	17.19	0.59	3.55%	145.84
2	http://www.alfikralarabi.org/	6492	3854	2638	17.4	17.9	0.5	2.87%	147.37
3	http://mentouri.ibda3.org/	5651	3363	2288	16.6	17.6	1	6.02%	130.00
4	http://alamatonline.net/	7342	4352	2990	17.9	19.1	1.2	6.70%	156.54
5	http://www.sawaleif.com	2382	1479	903	14.8	15.3	0.5	3.38%	59.02
6	http://www.ahram.org.eg/	4272	2444	1828	15.7	16.5	0.8	5.10%	110.79
7	http://www.alquds.co.uk/	1661	1006	655	14.3	14.8	0.5	3.50%	44.26
8	http://www.alriyadh.com/	6512	3951	2561	17.5	18.4	0.9	5.14%	139.18

6. References

[1] Aelphaeis Mangarae "Steganography FAQ," Zone-H.Org March 18th 2006

[2] S. Dickman, "An Overview of Steganography," July 2007.

[3] V. Potdar, E. Chang. "Visibly Invisible: Ciphertext as a Steganographic Carrier," Proceedings of *the 4th International Network Conference (INC2004), page(s):385–391, Plymouth, U.K., July 6–9, 2004*

[4] M. Al-Husainy "Image Steganography by Mapping Pixels to Letters," *2009 Science Publications*

[5] M. Shahreza, S. Shahreza, "Steganography in TeX Documents," *Proceedings of Intelligent System and Knowledge Engineering, ISKE 2008. 3rd International Conference, Nov. 2008*

[6] M. S. Shahreza, M. H. Shahreza, "An Improved Version of Persian/Arabic Text Steganography Using "La" Word" Proceedings *of IEEE 2008 6th National Conference on Telecommunication Technologies.*

[7]M. H. Shahreza, M. S. Shahreza, "A New Approach to Persian/Arabic Text Steganography " Proceedings *of 5th IEEE/ACIS International Conference on Computer and Information Science 2006*

[8] M. Aabed, S. Awaideh, A. Elshafei and A. Gutub "ARABIC DIACRITICS BASED STEGANOGRAPHY" Proceedings *of IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*

[9] W. Bender ,D. Gruhl ,N. Morimoto ,A. Lu "Techniques for data Hiding" Proceedings *OF IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996*

[10] K. Bennett, "Linguistic Steganography : survey, analysis, and robustness concerns for hiding information in text" Center for Education and Research in Information Assurance and Security, Purdue University 2004

[11] M. Nosrati , R. Karimi and, M. Hariri ," An introduction to steganography methods" *World Applied Programming, Vol (1), No (3), August 2011. 191-195.*

[12] M.H. Shirali-Shahreza, M. Shirali-Shahreza, " Text Steganography in chat" Proceedings of *3rd IEEE/IFIP International Conference in Central Asia on Sept. 2007*

[13] Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abdulelah Bin Mahfoodh " Improved Method of Arabic Text Steganography Using *the Extension „Kashida" Character" Bahria University Journal of Information & Communication Technology Vol.3, Issue 1, December 2010*

[14] A. Gutub, L. Ghouti, A. Amin, T. Alkharobi, M. Ibrahim. "Utilizing Extension Character Kashida with Pointed Letters for Arabic Text Digital Watermarking". Proceedings *of the International Conference on Security and Cryptography, Barcelona, Spain, July 28-13, 2007, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications. pages 329-332, INSTICC Press, 2007*

[15] Adnan Abdul-Aziz Gutub, and Manal Mohammad Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions " *World Academy of Science, Engineering and Technology 27 200*

[16] H. Shahreza, M. Shahreza "STEGANOGRAPHY IN PERSIAN AND ARABIC UNICODE TEXTS USING PSEUDO-SPACE AND PSEUDO CONNECTION CHARACTERS". *Journal of Theoretical and Applied Information Technology.*

[17] M. Bensaad, M. Yagoubi "High Capacity Diacritics-based Method For Information Hiding in Arabic Text" *2011 International Conference on Innovations in Information Technology.*

[18] A. Azmi and A. Alsaiari "Arabic Typography: A Survey" *International Journal of Electrical & Computer Sciences IJECS Vol: 9 No: 10*